

# Joint Tactical Radio System (JTRS) Standard Device Packet Signals Application Program Interface (API)



**Version: 1.2.2**  
**02 April 2008**

Statement A- Approved for public release; distribution is unlimited (29 March 2007)

**REVISION HISTORY**

| <b>Version</b> | <b>Authorization</b> | <b>Description</b>  | <b>Last Modified Date</b> |
|----------------|----------------------|---|---------------------------|
| 1.0            |                      | Initial release<br><b>ICWG Approved</b>   | 23-January-2006           |
| 1.1            |                      | Update outline format<br><b>ICWG Approved</b>   | 02-February-2006          |
| 1.2            |                      | Revised signalFlowResume()<br>text<br><b>ICWG Approved</b>                                    | 22-February-2006          |
| 1.2.1          |                      | Preparation for public release  | 29-March-2007             |
| 1.2.2          |                      | Errata: In<br>DevicePacketSignals.idl added<br>#ifndef/#define wrapper around<br>CF reference | 02-April-2008             |

## Table of Contents

|                                      |          |
|--------------------------------------|----------|
| <b>A. DEVICE PACKET SIGNALS.....</b> | <b>6</b> |
|--------------------------------------|----------|

---

## Table of Contents

|  |           |
|--|-----------|
| <b>A. DEVICE PACKET SIGNALS.....</b>                 | <b>6</b>  |
| <b>A.1 Introduction .....</b>                        | <b>6</b>  |
| A.1.1 Overview .....                                 | 6         |
| A.1.2 Service Layer Description .....                | 6         |
| A.1.3 Modes of Service .....                         | 6         |
| A.1.4 Service States .....                           | 6         |
| A.1.5 Referenced Documents .....                     | 7         |
| A.1.5.1 Government Documents .....                   | 7         |
| A.1.5.2 Commercial Standards .....                   | 7         |
| <b>A.2 Services .....</b>                            | <b>8</b>  |
| A.2.1 Provide Services .....                         | 8         |
| A.2.2 Use Services .....                             | 8         |
| A.2.3 Interface Modules .....                        | 8         |
| A.2.3.1 DevPktSig .....                              | 8         |
| A.2.4 Sequence Diagrams .....                        | 8         |
| <b>A.3 Service Primitives and Attributes.....</b>    | <b>9</b>  |
| A.3.1 DevPktSig::DevicePacketSignals .....           | 10        |
| A.3.1.1 <i>setMaxPayloadSize</i> Operation .....     | 10        |
| A.3.1.2 <i>setMinPayloadSize</i> Operation .....     | 11        |
| A.3.1.3 <i>setDesiredPayloadSize</i> Operation ..... | 12        |
| A.3.1.4 <i>setMinOverrideTimeout</i> Operation.....  | 13        |
| A.3.1.5 <i>signalEmpty</i> Operation .....           | 14        |
| A.3.1.6 <i>signalFlowResume</i> Operation .....      | 15        |
| <b>A.4 IDL.....</b>                                  | <b>16</b> |
| A.4.1 DevicePacketSignals IDL.....                   | 16        |
| <b>A.5 UML .....</b>                                 | <b>17</b> |
| <b>Appendix A.A Abbreviations and Acronyms .....</b> | <b>18</b> |
| <b>Appendix A.B Performance Specification.....</b>   | <b>19</b> |

## **Lists of Figures**

FIGURE 1 – DEVICEPACKETSIGNALS INTERFACE CLASS DIAGRAM..... 8

## A. DEVICE PACKET SIGNALS

### A.1 INTRODUCTION

This document defines a common set of *Device Packet Signals* interfaces to be used by Joint Tactical Radio (JTR) Set Applications and Services. The *Device Packet Signals* interface provides the ability to receive the setting of control signals, payload sizes, and number of priority queues for packet data flow.

The *Device Packet Signals* interfaces are documented within to minimize coupling between the device and service interfaces that utilize the *Device Packet Signals* interfaces.

#### A.1.1 Overview

- a. Section A.1, *Introduction*, of this document contains the introductory material regarding the overview and referenced documents of this document.
- b. Section A.2, *Services*, provides a summary of service interface uses and sequence diagrams.
- c. Section A.3, *Service Primitives and Attributes*, specifies the operations that are provided by the *Device Packet Signals* interface.
- d. Section A.4, *IDL*.
- e. Section A.5, *UML*.
- f. Appendix A.A, *Abbreviations and Acronyms*.
- g. Appendix A.B, *Performance Specification*.

#### A.1.2 Service Layer Description

Not applicable.

#### A.1.3 Modes of Service

Not applicable.

#### A.1.4 Service States

Not applicable.

## **A.1.5 Referenced Documents**

The following documents of the exact issue shown form a part of this specification to the extent specified herein.

### **A.1.5.1 Government Documents**

#### **A.1.5.1.1 Specifications**

##### **A.1.5.1.1.1 Federal Specifications**

None

##### **A.1.5.1.1.2 Military Specifications**

None

#### **A.1.5.1.2 Other Government Agency Documents**

[1] JTRS Standard, "Software Communications Architecture (SCA)," JPEO, Version 2.2.2.

### **A.1.5.2 Commercial Standards**

None

## A.2 SERVICES

### A.2.1 Provide Services

Not applicable.

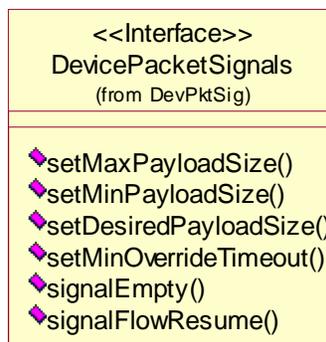
### A.2.2 Use Services

Not applicable.

### A.2.3 Interface Modules

#### A.2.3.1 DevPktSig

##### A.2.3.1.1 DevicePacketSignals Interface Description



**Figure 1 – DevicePacketSignals Interface Class Diagram**

The interface design of *DevicePacketSignals* is shown in Figure 1. *DevicePacketSignals* interface provides the ability to signal flow control to the packet producer that is derived from this interface.

### A.2.4 Sequence Diagrams

None

## **A.3 SERVICE PRIMITIVES AND ATTRIBUTES**

To enhance the readability of this API document and to avoid duplication of data, the type definitions of all structured types (i.e., structures, typedefs, exceptions, enumerations and unions) used by the Service Primitives and Attributes have been co-located in section A.5. This cross-reference of types also includes any nested structures in the event of a structure of structures or an array of structures.

## A.3.1 DevPktSig::DevicePacketSignals

### A.3.1.1 *setMaxPayloadSize* Operation

The *setMaxPayloadSize* operation sets the absolute maximum payload size allowed for a payload passed to the *pushPacket* operation.

#### A.3.1.1.1 Synopsis

```
void setMaxPayloadSize(in unsigned long maxPayloadSize);
```

#### A.3.1.1.2 Parameters

| Parameter Name | Description                                     | Type          | Units |
|----------------|---|---------------|-------|
| maxPayloadSize | The maximum payload size allowed for a payload. | unsigned long | bytes |

#### A.3.1.1.3 State

Not applicable.

#### A.3.1.1.4 New State

Not applicable.

#### A.3.1.1.5 Return Value

None

#### A.3.1.1.6 Originator

Not applicable.

#### A.3.1.1.7 Exceptions

| Exception  | Attributes        | Type                            | Description  |
|--|-------------------|---------------------------------|--|
| CF::PropertySet::InvalidConfiguration<br>(see SCA [1]) | invalidProperties | CF::Properties<br>(see SCA [1]) | Sequence of invalid properties.                                      |
|  | msg               | string                          | A message of type string indicating that the exception has occurred. |

### A.3.1.2 *setMinPayloadSize* Operation

The *setMinPayloadSize* operation is used for asynchronous modes. It sets the minimum payload size allowed for a payload passed to the *pushPacket* operation. Note that payloads of 0 (zero) size (i.e. control packet) are exempt.

#### A.3.1.2.1 Synopsis

*void setMinPayloadSize(in unsigned long minPayloadSize);*

#### A.3.1.2.2 Parameters

| Parameter Name | Description                                     | Type          | Units |
|----------------|---|---------------|-------|
| minPayloadSize | The minimum payload size allowed for a payload. | unsigned long | bytes |

#### A.3.1.2.3 State

Not applicable.

#### A.3.1.2.4 New State

Not applicable.

#### A.3.1.2.5 Return Value

None

#### A.3.1.2.6 Originator

Not applicable.

#### A.3.1.2.7 Exceptions

None

### A.3.1.3 *setDesiredPayloadSize* Operation

The *setDesiredPayloadSize* operation is used for synchronous modes. It sets the desired payload size allowed for a payload passed to the *pushPacket* operation.

#### A.3.1.3.1 Synopsis

*void setDesiredPayloadSize(in unsigned long desiredPayloadSize);*

#### A.3.1.3.2 Parameters

| Parameter Name     | Description                                     | Type          | Units |
|--------------------|---|---------------|-------|
| desiredPayloadSize | The desired payload size allowed for a payload. | unsigned long | bytes |

#### A.3.1.3.3 State

Not applicable.

#### A.3.1.3.4 New State

Not applicable.

#### A.3.1.3.5 Return Value

None

#### A.3.1.3.6 Originator

Not applicable.

#### A.3.1.3.7 Exceptions

| Exception  | Attributes        | Type                            | Description  |
|--|-------------------|---------------------------------|--|
| CF::PropertySet::InvalidConfiguration<br>(see SCA [1]) | invalidProperties | CF::Properties<br>(see SCA [1]) | Sequence of invalid properties.                                      |
|  | msg               | string                          | A message of type string indicating that the exception has occurred. |

### A.3.1.4 *setMinOverrideTimeout* Operation

The *setMinOverrideTimeout* operation sets the time a payload smaller than the “minPayloadSize” for an asynchronous mode or “desiredPayloadSize” for a synchronous mode, should be held before passed to the *pushPacket* operation.

#### A.3.1.4.1 Synopsis

*void setMinOverrideTimeout(in unsigned long minOverrideTimeout);*

#### A.3.1.4.2 Parameters

| Parameter Name     | Description                                 | Type          | Units |
|--------------------|---|---------------|-------|
| minOverrideTimeout | The minimum timeout value to be overridden. | unsigned long | ms    |

#### A.3.1.4.3 State

Not applicable.

#### A.3.1.4.4 New State

Not applicable.

#### A.3.1.4.5 Return Value

None

#### A.3.1.4.6 Originator

Not applicable.

#### A.3.1.4.7 Exceptions

| Exception  | Attributes        | Type                            | Description  |
|--|-------------------|---------------------------------|--|
| CF::PropertySet::InvalidConfiguration<br>(see SCA [1]) | invalidProperties | CF::Properties<br>(see SCA [1]) | Sequence of invalid properties.                                      |
|  | msg               | string                          | A message of type string indicating that the exception has occurred. |

### **A.3.1.5 *signalEmpty* Operation**

The *signalEmpty* operation is used to signal an empty condition to the packet producer when the specified stream has been emptied after an *endOfStream* condition.

#### **A.3.1.5.1 Synopsis**

*oneway void signalEmpty(in unsigned long streamId);*

#### **A.3.1.5.2 Parameters**

| <b>Parameter Name</b> | <b>Description</b>                    | <b>Type</b>   | <b>Units</b> |
|-----------------------|---------------------------------------|---------------|--------------|
| streamId              | Identification number for the stream. | unsigned long | bytes        |

#### **A.3.1.5.3 State**

Not applicable.

#### **A.3.1.5.4 New State**

Not applicable.

#### **A.3.1.5.5 Return Value**

None

#### **A.3.1.5.6 Originator**

Not applicable.

#### **A.3.1.5.7 Exceptions**

None

### **A.3.1.6 *signalFlowResume* Operation**

The *signalFlowResume* operation is used to signal the packet producer when the consumer (i.e. *Device User*) has transitioned from a “no space available” in the specified priority queue to a “space available” condition.

#### **A.3.1.6.1 Synopsis**

*oneway void signalFlowResume(in octet priorityQueueID);*

#### **A.3.1.6.2 Parameters**

| <b>Parameter Name</b> | <b>Description</b>           | <b>Type</b> | <b>Units</b> |
|-----------------------|------------------------------|-------------|--------------|
| priorityQueueID       | A priority queue ID in octet | octet       | octet        |

#### **A.3.1.6.3 State**

Not applicable.

#### **A.3.1.6.4 New State**

Not applicable.

#### **A.3.1.6.5 Return Value**

None

#### **A.3.1.6.6 Originator**

Not applicable.

#### **A.3.1.6.7 Exceptions**

None

## A.4 IDL

### A.4.1 DevicePacketSignals IDL

```
/*
** DevicePacketSignals.idl
**/

#ifndef __DEVICEPACKETSIGNALS_DEFINED
#define __DEVICEPACKETSIGNALS_DEFINED

#ifndef __CF_DEFINED
#include "CF.idl"
#endif

/* DevPktSig */

module DevPktSig {

    interface DevicePacketSignals {

        void setMaxPayloadSize (
            in unsigned long maxPayloadSize
        )
        raises (CF::PropertySet::InvalidConfiguration);

        void setMinPayloadSize (
            in unsigned long minPayloadSize
        )
        raises (CF::PropertySet::InvalidConfiguration);

        void setDesiredPayloadSize (
            in unsigned long desiredPayloadSize
        )
        raises (CF::PropertySet::InvalidConfiguration);

        void setMinOverrideTimeout (
            in unsigned long minOverrideTimeout
        )
        raises (CF::PropertySet::InvalidConfiguration);

        oneway void signalEmpty ();

        oneway void signalFlowResume (
            in octet priorityQueueID
        );

    };

};

#endif
```

## **A.5 UML**

Not Applicable.

## **APPENDIX A.A ABBREVIATIONS AND ACRONYMS**

|             |                                      |
|-------------|--------------------------------------|
| <b>API</b>  | Application Program Interface        |
| <b>CF</b>   | Core Framework                       |
| <b>ICWG</b> | Interface Control Working Group      |
| <b>ID</b>   | IDentifier                           |
| <b>IDL</b>  | Interface Definition Language        |
| <b>JPEO</b> | Joint Program Executive Office       |
| <b>JTRS</b> | Joint Tactical Radio System          |
| <b>ms</b>   | millisecond                          |
| <b>SCA</b>  | Software Communications Architecture |
| <b>UML</b>  | Unified Modeling Language            |

## **APPENDIX A.B PERFORMANCE SPECIFICATION**

Not applicable.